

KEKER & VAN NEST LLP
 ROBERT A. VAN NEST - # 84065
 rvannest@kvn.com
 CHRISTA M. ANDERSON - # 184325
 canderson@kvn.com
 DANIEL PURCELL - # 191424
 dpurcell@kvn.com
 633 Battery Street
 San Francisco, CA 94111-1809
 Telephone: 415 391 5400
 Facsimile: 415 397 7188

KING & SPALDING LLP
 DONALD F. ZIMMER, JR. - #112279
 fzimmer@kslaw.com
 CHERYL A. SABNIS - #224323
 csabnis@kslaw.com
 101 Second Street, Suite 2300
 San Francisco, CA 94105
 Tel: 415.318.1200
 Fax: 415.318.1300

KING & SPALDING LLP
 SCOTT T. WEINGAERTNER
 (*Pro Hac Vice*)
 sweingaertner@kslaw.com
 ROBERT F. PERRY
 rperry@kslaw.com
 BRUCE W. BABER (*Pro Hac Vice*)
 1185 Avenue of the Americas
 New York, NY 10036
 Tel: 212.556.2100
 Fax: 212.556.2222

IAN C. BALLON - #141819
 ballon@gtlaw.com
 HEATHER MEEKER - #172148
 meekerh@gtlaw.com
 GREENBERG TRAURIG, LLP
 1900 University Avenue
 East Palo Alto, CA 94303
 Tel: 650.328.8500
 Fax: 650.328.8508

Attorneys for Defendant
 GOOGLE INC.

UNITED STATES DISTRICT COURT
 NORTHERN DISTRICT OF CALIFORNIA
 SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.,

Plaintiff,

v.

GOOGLE INC.,

Defendant.

Case No. 3:10-cv-03561 WHA

GOOGLE INC.'S OPPOSITION TO
 ORACLE AMERICA, INC.'S MOTION FOR
 JUDGMENT AS A MATTER OF LAW
 UNDER RULE 50(b) OR, IN THE
 ALTERNATIVE, FOR A NEW TRIAL

Date: July 26, 2012
 Time: 8:00 a.m.
 Dept.: Courtroom 8, 19th Floor
 Judge: Hon. William Alsup

TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	1
ARGUMENT	1
I. ORACLE HAS FAILED TO ESTABLISH THAT, AS A MATTER OF LAW, GOOGLE INFRINGED ORACLE’S COPYRIGHTS.....	1
A. Oracle did not prove ownership of the eleven “copied” files, which were not registered with the Copyright Office.	1
B. The jury properly found that any copying of code comments was <i>de</i> <i>minimis</i>	2
C. The jury properly found that Google did not infringe Oracle’s copyrights in the Java specifications.....	3
D. A reasonable jury could find that Google made a fair use of the SSO of the 37 Java API packages.	4
1. Oracle’s motion is moot.....	4
2. A reasonable jury could find that the purpose and character of Google’s use of the SSO of the 37 API packages favors fair use.....	5
3. A reasonable jury could find that the nature of the SSO of the 37 API packages is highly functional.	8
4. A reasonable jury could find that Google used only so much of the J2SE API packages as is necessary for compatibility.....	9
5. A reasonable jury could find that there has been no adverse impact on the actual or potential market for Oracle’s work.	10
E. Oracle is not entitled to judgment as a matter of law on copyrightability.....	12
1. Oracle’s motion is improper.	12
2. The Court correctly concluded that the SSO of the API packages is not copyrightable.	13
F. Oracle is not entitled to judgment as a matter of law on derivative work.	16
II. THE JURY PROPERLY FOUND THAT GOOGLE DID NOT INFRINGE THE ASSERTED CLAIMS OF THE 104 PATENT.	17
A. Dalvik bytecode instructions only contain indexes, which are not “symbolic references” because they refer to numeric memory locations.....	17
1. Field indexes are numeric references, not symbolic references.	17

1	B.	The Court’s claim construction applies to data in the constant pool tables as well as the actual field data in an instance object.	18
2	C.	Indexes are not symbolic references because they are not resolved.	19
3	D.	Dexopt does not infringe the ’104 patent.	20
4	III.	THE JURY PROPERLY FOUND THAT THE ANDROID DX TOOL DID NOT INFRINGE THE ASSERTED CLAIMS OF THE ’520 PATENT.	20
5	IV.	ORACLE IS NOT ENTITLED TO JUDGMENT AS A MATTER OF LAW ON GOOGLE’S EQUITABLE DEFENSES	22
6	A.	Oracle’s motion is improper.	22
7	B.	Oracle’s motion is moot.	23
8	C.	Oracle’s substantive arguments also lack merit.	23
9	1.	There is sufficient evidence to support a finding that Oracle’s claims are barred by equitable estoppel.	23
10	2.	Oracle’s claims are barred by laches.	24
11	V.	ALTERNATIVE DEFENSES TO PATENT INFRINGEMENT	25
12	VI.	ORACLE IS NOT ENTITLED TO A NEW TRIAL	25
13	CONCLUSION	25

TABLE OF AUTHORITIES

Page(s)**Federal Cases**

<i>Allen v. Academic Games League of Am., Inc.</i> 89 F.3d 614 (9th Cir. 1996)	14
<i>Apple Computer, Inc. v. Microsoft Corp.</i> 35 F.3d 1435	3
<i>ATS Prods. Inc. v. Ghiorso</i> No. C10-4880, 2012 WL 1067547 (N.D. Cal. Mar. 28, 2012).....	13, 23
<i>Baker v. Selden</i> 101 U.S. 99 (1879).....	14
<i>Bateman v. Mnemonics, Inc.</i> 79 F.3d 1532 (11th Cir. 1996)	8
<i>Boisson v. Banian, Ltd.</i> 273 F.3d 262 (2d Cir. 2001).....	2
<i>Campbell v. Acuff-Rose Music, Inc.</i> 510 U.S. 569 (1994).....	5, 7, 10
<i>Computer Assocs. Int'l, Inc. v. Altai, Inc.</i> 982 F.2d 693 (2d Cir. 1992).....	14
<i>Corp. of Am. v. Universal City Studios, Inc.</i> 464 U.S. 417 (1984).....	10
<i>Eon-Net LP v. Flagstar Bancorp</i> 653 F.3d 1314 (Fed. Cir. 2011).....	22
<i>Fisher v. Dees</i> 794 F.2d 432 (9th Cir. 1986)	2
<i>Gates Rubber Co. v. Bando Chem. Indus., Ltd.</i> 9 F.3d 823 (10th Cir. 1993)	14, 15
<i>Granite State Ins. Co. v. Smart Modular Techs., Inc.</i> 76 F.3d 1023 (9th Cir. 1996)	2, 12, 23
<i>Herbert Rosenthal Jewelry Corp. v. Kalpakian</i> 446 F.2d 738 (9th Cir. 1971)	14
<i>In re Katz Interactive Call Processing Patent Litig.</i> 712 F. Supp. 2d 1080 (C.D. Cal. 2010)	24
<i>Incredible Techs., Inc. v. Virtual Techs., Inc.</i> 400 F.3d 1007 (7th Cir. 2005)	3

1	<i>Kelly v. Arriba Soft Corp.</i>	
2	336 F.3d 811 (9th Cir. 2003)	9, 10
3	<i>Lotus Dev. Corp. v. Borland Int'l, Inc.</i>	
4	49 F.3d 807 (1995) (Boudin, J., concurring), <i>aff'd by an equally divided court</i> ,	
5	516 U.S. 233 (1996).....	8
6	<i>Lucent Techs., Inc. v. Gateway, Inc.</i>	
7	580 F. Supp. 2d 1016 (S.D. Cal. 2008) <i>aff'd in part, vacated in part on other</i>	
8	<i>grounds, remanded</i> , 580 F.3d 1301 (Fed. Cir. 2009)	24
9	<i>Micro Star v. Formgen Inc.</i>	
10	154 F.3d 1107 (9th Cir. 1998)	17
11	<i>Mirage Editions v. Albuquerque A.R.T. Co.</i>	
12	856 F.2d 1341 (9th Cir. 1988)	16, 17, 18, 20
13	<i>Molski v. M.J. Cable, Inc.</i>	
14	481 F.3d 724 (9th Cir. 2007)	25
15	<i>Newton v. Diamond</i>	
16	388 F.3d 1189 (9th Cir. 2004)	2
17	<i>Sega Enters. Ltd. v. Accolade, Inc.</i>	
18	977 F.2d 1510 (9th Cir. 1992)	8, 9, 15
19	<i>Sheldon L. Pollack Corp. v. Universal Health Servs. Inc.</i>	
20	1990 WL 194282 (9th Cir. Dec. 6, 1990).....	13
21	<i>Sony Computer Enter. v. Connectix Corp.</i>	
22	203 F.3d 596 (9th Cir. 2000)	<i>passim</i>
23	<i>Swirsky v. Carey</i>	
24	376 F.3d 841 (9th Cir. 2004)	14
25	<u>Federal Statutes</u>	
26	17 U.S.C. § 102.....	4
27	17 U.S.C. § 102(b)	8, 15, 16
28	17 U.S.C. § 103(b)	2
	<u>Federal Rules</u>	
	Fed. R. Civ. P. 50.....	2, 12, 21, 23
	Fed. R. Civ. P. 50(a)	24, 25
	Fed. R. Civ. P. 50(b)	3, 13, 14, 25
	Fed. R. Civ. P. 52(b)	12, 13, 23
	Fed. R. Civ. P. 59.....	13

INTRODUCTION

The Court should deny Oracle's renewed motion for judgment as a matter of law ("JMOL"), or, in the alternative, a new trial. Oracle's motion relies on conclusions from disputed evidence that are not taken in the light most favorable to the jury's verdict, ignores evidence supporting the jury's verdict, addresses legal issues that were never put to the jury in the first place, and asks the Court to rule on several issues that are now moot. It should be denied.

ARGUMENT

I. ORACLE HAS FAILED TO ESTABLISH THAT, AS A MATTER OF LAW, GOOGLE INFRINGED ORACLE'S COPYRIGHTS.

A. Oracle did not prove ownership of the eleven "copied" files, which were not registered with the Copyright Office.

Oracle's JMOL motion as to ownership of "the asserted copyrights" is improper, without any basis, and unnecessary. In its May 16 Order denying Google's JMOL motion regarding ownership and registration (Dkt. 1165 at 2), the Court found that Google had the burden of proof on ownership of the copyrights in the works on which Oracle based its claim of infringement and that the presumption of ownership applied even as to individual portions of the copyrighted works. While Google believes the Court's ruling was incorrect, there is no need for a further order on the copyright ownership issue because the underlying liability issues have been resolved.

There is, moreover, no basis in the record for the ruling Oracle seeks. Oracle argues that it proved it is the "owner of the copyrighted works," seeks a ruling that it is the owner of "the asserted works" and cites again case law relating to the presumptions flowing from a copyright registration. Dkt. 1212 at 2-3. Yet Oracle overlooks that Google has made clear that Google does not contest Oracle's ownership of the copyrights on which Oracle sued Google, i.e., the copyrights in the Java SE 1.4 and 5.0 platforms. *See* Dkt. 1107 at 3. Google raised a continuing issue of ownership only because Oracle persuaded the Court (over Google's objections) that it could submit to the jury infringement "claims" based on the eleven individual files—less than the entire platform works in which the copyrights were registered. The copyrights in those individual files were *not* separately registered with the Copyright Office, and no presumption therefore

1 applies to them. Oracle failed to introduce any evidence that Oracle was the author of or the
 2 owner of the copyright rights in any of those eleven individual files; there was simply no
 3 testimony from any witness on that issue and therefore no issue for the jury to decide.¹ *See, e.g.,*
 4 *Boisson v. Banian, Ltd.*, 273 F.3d 262, 268 (2d Cir. 2001); *see also* 17 U.S.C. § 103(b).

5 For these reasons and those argued by Google in its prior submissions on this issues
 6 (which Google incorporates herein by reference), Oracle is not entitled to a ruling that it proved
 7 ownership of the “asserted copyrights.” *See, e.g.,* Dkt. 1092 at 1; Dkt. 1079 at 1-2, 4, 47-49; Dkt.
 8 1043 at 18; Dkt. 1007 at 1.

9 **B. The jury properly found that any copying of code comments was *de minimis*.**

10 “For an unauthorized use of a copyrighted work to be actionable, the use must be
 11 significant enough to constitute infringement. This means that even where the fact of copying is
 12 conceded, no legal consequences will follow from that fact unless the copying is substantial.”
 13 *Newton v. Diamond*, 388 F.3d 1189, 1192-93 (9th Cir. 2004) (internal citations omitted).
 14 “Substantiality is measured by considering the qualitative and quantitative significance of the
 15 copied portion in relation to the plaintiff’s work as a whole.” *Id.* at 1195.

16 The evidence at trial showed that the comments are not compiled into object code and do
 17 not appear on phones. RT 1317:9-20, 1318:4-6 (Mitchell). Oracle’s own expert conceded that
 18 the comments “don’t have any impact, whatsoever, in how the program runs once it’s compiled.”
 19 *See* RT 1318:1-3 (Mitchell). Based on this testimony alone, a reasonable jury could find that the
 20 comments were qualitatively insignificant, and Oracle’s motion should be denied. *See Newton*,
 21 388 F.3d at 1995 (significance of copying should be “measured by considering the qualitative and
 22 quantitative significance of the copied portion.”).

23 Oracle also failed to present evidence showing that the “average audience” would
 24 recognize the alleged copying of the source code comments. *See Fisher v. Dees*, 794 F.2d 432,
 25 434 n.2 (9th Cir. 1986). Because there is no evidence that the code comments are anything other

26
 27 ¹ This issue was decided by the Court as a matter of law, and Rule 50 does not apply to issues
 28 decided by the Court. *Granite State Ins. Co. v. Smart Modular Techs., Inc.*, 76 F.3d 1023, 1030-
 31 (9th Cir. 1996).

1 than trivial and insignificant, the jury properly found that any copying was *de minimis*.

2 **C. The jury properly found that Google did not infringe Oracle’s copyrights in**
 3 **the Java specifications.**

4 Oracle argues that a reasonable jury would be compelled to find that Google infringed the
 5 SSO of its English-language documentation. Mot. at 4-5. This theory was never put to the jury,
 6 and is therefore not properly the subject of a Rule 50(b) motion. In any event, the SSO of the
 7 Java specifications is not copyrightable expression for the same reasons that the SSO of the 37
 8 Java API packages is not copyrightable expression. *See* Part I.E, *infra*. Indeed, the English-
 9 language documentation is created automatically by a computer tool (“javadoc”) from the APIs
 10 that are being documented. *See* RT 1168:21 – 1169:15 (Lee), RT 607:18-24, 614:1-4 (Reinhold).
 11 Because the documentation’s SSO is always derived from the APIs being documented, there is no
 12 originality or creativity in the SSO of the documentation independent of the SSO of the 37 API
 13 packages. Any similarities between the Java and Android documentation are a result of both
 14 platforms using the same unprotectable names and functional organization. RT 541:21-542:10
 15 (Screven); *see also* Dkt. 1202 at 40-41. Oracle therefore is not entitled to JMOL on the alleged
 16 infringement of the SSO of its Java API documentation.

17 Setting aside the SSO of the documentation, the jury properly found that the English-
 18 language descriptions in the Android documentation were not “virtually identical” to Oracle’s
 19 Java documentation. Oracle argues that the Court should have instructed the jury to apply the
 20 “substantial similarity” test to the allegedly copied documentation. Mot. at 5. Oracle is wrong.
 21 The documentation is a factual description of underlying functionality. RT 1175:10-14 (Lee); RT
 22 2204:1-15 (Astrachan). Protection for factual or functional works is thin. *See Apple Computer,*
 23 *Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1439; *see also Incredible Techs., Inc. v. Virtual Techs.,*
 24 *Inc.*, 400 F.3d 1007, 1013 (7th Cir. 2005) (“utilitarian explanations” of a system “are not
 25 sufficiently original or creative to merit copyright protection,” or alternatively are protected “only
 26 against virtually identical copying”). The virtual identity standard is the correct one.

27 Even if substantial similarity were the correct standard—and it is not—a reasonable jury
 28 still could find that Android’s documentation did not infringe Oracle’s copyrights. At trial,

Oracle presented to the jury only three snippets of allegedly copied documentation. The three snippets were sufficiently different that a reasonable jury could easily find that they were not substantially similar, much less virtually identical. *Compare* TX 610.2 with TX 767; *see* also RT 1169:25-1170:19 (Lee) (comparing descriptions of the CipherInputStream class), 1171:3-1172:25 (comparing descriptions of the Cipher class), 1174:17-1175:9 (comparing descriptions of the Pipe class). Moreover, the exhibits from which those three snippets came included many other even more dissimilar samples of documentation, samples that Oracle does not address in its JMOL. *Compare* TX 610.2 with TX 767; RT 1326:10-1328:23 (Mitchell) (discussing “KeyPair” method). Even if those examples were substantially similar or virtually identical to each other as matter of law—they are not—they constituted only a handful of sentences, out of thousands of pages of documentation. RT 617-2:7 (Reinhold). Under these circumstances, a reasonable jury could find that Oracle failed to carry its burden of proving that the Android documentation, taken as a whole, infringed Oracle’s Java documentation, as a whole.²

D. A reasonable jury could find that Google made a fair use of the SSO of the 37 Java API packages.

A reasonable jury could find that all four of the statutory fair use factors favor Google, or at least that three factors favor Google while other is neutral. Weighing these factors together, a reasonable jury could conclude that Google made a fair use of the SSO of the 37 API packages.

1. Oracle’s motion is moot.

Oracle’s motion should be denied as moot. The Court has ruled that the SSO of the 37 API packages is not copyrightable. Dkt. 1202. Google therefore was free to use the SSO of the 37 API packages, regardless of whether a reasonable jury could conclude that Google’s use was also a fair use under the Copyright Act. Accordingly, Oracle’s motion should be denied as moot.

² Additionally, a reasonable jury could find that any similarities between the Android and Java documentation is a result of the fact that two sets of documentation describe the same ideas and functionality. As Mr. Lee noted when asked about documentation of the Pipe class in the java.nio.channels package, “they contain the same words, certainly, but that’s to be expected when you’re trying to – you’re describing various specific concepts in as few words as possible.” RT 1175:10-24 (Lee). When asked about similarity, Prof. Mitchell also stated that the examples “tell you the same information” and “mean the same thing.” RT 1328:2-4, 9-10 (Mitchell). A reasonable jury could therefore conclude that any similarities in the documentation are similar ideas, not similar expressions. That is not copyright infringement. 17 U.S.C. § 102.

1 **2. A reasonable jury could find that the purpose and character of**
 2 **Google's use of the SSO of the 37 API packages favors fair use.**

3 As discussed below, Google's use of the SSO of the 37 API packages should not be
 4 characterized as a wholly "commercial" use. But even if it is so characterized, "the language of
 5 the statute makes clear that the commercial or nonprofit educational purpose of a work is only
 6 one element of the first factor enquiry into its purpose and character." *Campbell v. Acuff-Rose*
 7 *Music, Inc.*, 510 U.S. 569, 584 (1994). "Congress resisted attempts to narrow the ambit of this
 8 traditional enquiry by adopting categories of presumptively fair use, and it urged courts to
 9 preserve *the breadth of their traditionally ample view of the universe of relevant evidence.*" *Id.*
 10 (citing *Harper & Row, Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 561 (1985)) (emphasis
 11 added). "Accordingly, the mere fact that a use is educational and not for profit does not insulate
 12 it from a finding of infringement, *any more than the commercial character of a use bars a finding*
 13 *of fairness.*" *Campbell*, 510 U.S. at 584 (emphasis added). Indeed, most of the exemplary fair
 14 uses listed in the preamble to section 107 are "generally conducted for profit in this country." *Id.*
 15 (quotation marks and citation omitted). The Ninth Circuit has recognized that, after *Campbell*, it
 16 is improper to apply a presumption against fair use based on a defendant's commercial purpose.
 17 *Sony Computer Enter. v. Connectix Corp.*, 203 F.3d 596, 606 (9th Cir. 2000). Instead, a
 18 commercial purpose "is only a 'separate factor that tends to weigh against a finding of fair use.'" *Id.*
 19 (quoting *Campbell*, 510 U.S. at 585).

20 But even a commercial use does not as a matter of law tilt the first factor against fair use.
 21 The more the purpose and character of the defendant's use is transformative, "the less will be the
 22 significance of other factors, like commercialism, that may weigh against a finding of fair use."
 23 *Campbell*, 510 U.S. at 579. A use is transformative where it "adds something new, with a further
 24 purpose or different character, altering the first with new expression, meaning, or message[.]" *Id.*
 25 (citing Pierre N. Leval, *Toward a Fair Use Standard*, 103 HARV. L. REV. 1105, 1111 (1990)).
 26 "[T]he goal of copyright, to promote science and the arts, is generally furthered by the creation of
 27 transformative works" *Campbell*, 510 U.S. at 579.

28 In *Sony v. Connectix*, the Ninth Circuit found that Connectix's Virtual Game Station

1 (“VGS”) software was a product that “creates a new platform, the personal computer, on which
 2 consumers can play games designed for the Sony PlayStation.” 203 F.3d at 606. Because VGS
 3 was an “innovation” that “affords opportunities for game play in new environments,” the Ninth
 4 Circuit found that VGS was “modestly transformative.” *Id.* The court so found “notwithstanding
 5 the similarity of uses and functions between the Sony PlayStation and the Virtual Game
 6 Station”—that is, notwithstanding that the very purpose of VGS was to allow users to play *the*
 7 *same games* that users play using the Sony PlayStation. The Ninth Circuit was “at a loss” to see
 8 “how Connectix’s drafting of entirely new object code for its VGS program could not be
 9 transformative, despite the similarities in function and screen output.” *Id.* at 606-07.

10 When Google created Android, it, too, created a new platform. Sun itself recognized that
 11 Android was innovative. *See* TX 435 (email from Schwartz to Schmidt stating, “Sun is ready
 12 embrace Google’s innovation”); TX 2352 (blog post from Schwartz congratulating Google’s
 13 “*new Java/Linux phone platform, Android.*”) (emphasis added). In contrast to J2SE, Android
 14 provides a “full stack” solution for mobile computing. RT 1938:10-1939:12 (Rizvi). Android
 15 builds on the SSO of the 37 API packages to provide developers with access to a full application
 16 framework suitable for smartphone applications written in the Java programming language. RT
 17 1682:23-1684:1 (Rubin). This work required approximately three years to complete. RT
 18 1684:20-24 (Rubin). Android created opportunities to use the APIs in the 37 packages in a new
 19 environment, namely the Android smartphone platform. RT 2182:3-7 (Astrachan).

20 And although there are similarities between Android and the J2SE platforms in the
 21 structure, sequence and organization of the API elements in the 37 packages, the *implementing*
 22 *code* in Android and J2SE is very (and undisputedly) different. RT 2182:25-2183:1 (Astrachan);
 23 *see also* RT 2184:8-21, 2185:10-2186:17 (Astrachan) (implementing code “completely
 24 different”); RT 2297:7-2299:13 (Mitchell) (agreeing code is different). In fact, the nine-line
 25 rangeCheck method is the only evidence of any similarities in the *implementing code* itself. *See*
 26 RT 1309:8-1313:11 (Mitchell); RT 2182:13-2183:1 (Astrachan). Thus, notwithstanding any
 27 similarities between how the SSO of the 37 API packages are used in Android and the J2SE
 28 platforms, a reasonable jury could find that Android is transformative. *See Sony v. Connectix,*

203 F.3d at 606-07. Oracle’s renewed motion does not address any of this evidence, based on which a reasonable jury could find that Android—like VGS—is sufficiently transformative to tilt the first statutory factor in favor of fair use.

Oracle argues that the defendant’s work must have an *entirely* different purpose than the plaintiff’s work in order to be transformative. Mot. at 7:9-12. This ignores the Supreme Court’s definition of “transformative” in *Campbell*. 510 U.S. at 579 (use is transformative where it “adds something new, with a further purpose or different character, altering the first with new expression, meaning, or message”); *see also* Dkt. 1018 at 13. In *Campbell*, the transformative use was a popular song recording that parodied the plaintiff’s popular song recording. While the original was a rock ballad and the other from the hip hop genre, the two works both were commercial entertainment, popular songs targeting mainstream audiences. Moreover, although the cases Oracle cites involved situations where the defendant’s work had a very different purpose than the plaintiff’s work, none of those cases hold that a work *cannot* be transformative if it has a less than entirely different purpose. Thus, none of those cases narrow *Campbell*—nor could they, given that *Campbell* is a Supreme Court decision that is binding on the Ninth Circuit.

Here, Android not only provided a further purpose and different character to the SSO of the 37 API packages, allowing them to be used in a new environment and platform, but also added over one hundred new API packages that interact with and inter-depend on those 37 packages, transforming the SSO of the API packages at issue into something new. RT 1680:24-1682:5, 1682:23-1684:1 (Rubin); TX 51 (list of API packages in Android version 2.1). And the purpose of implementing the API packages was to achieve compatibility with code written to use APIs from those 37 packages. RT 1782:6-17 (Bornstein); RT 2183:6-11 (Astrachan); RT 803:9-20 (Bloch); *see* Dkt. 1202 at 38 (“Google replicated what was necessary to achieve a degree of interoperability—but no more, taking care, as said before, to provide its own implementations.”).

To reach a final finding on the first factor, the *Sony v. Connectix* court weighed VGS’s transformative purpose against Connectix’s commercial use of Sony’s copyrighted work. Connectix had reverse engineered Sony’s code in order to determine its functional requirements. 203 F.3d at 601. Because the final VGS product did not incorporate Sony’s copyrighted code, the

1 court found that Connectix's commercial use was only indirect or derivative. *Id.* at 607.
 2 Moreover, the use was for the purpose of achieving compatibility, a legitimate use under the first
 3 factor. *Id.*; *see also Bateman v. Mnemonics, Inc.*, 79 F.3d 1532, 1547 (11th Cir. 1996) (external
 4 factors such as compatibility can negate a finding of infringement "per 17 U.S.C. § 102(b)" or "a
 5 finding of fair use, copyright estoppel, or misuse"); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49
 6 F.3d 807, 821 (1995) (Boudin, J., concurring) (suggesting Borland's use of Lotus's menu
 7 hierarchy, in addition to being noninfringing by virtue of 17 U.S.C. § 102(b), might also be a fair
 8 use), *aff'd by an equally divided court*, 516 U.S. 233 (1996). Weighing these facts together, the
 9 Ninth Circuit found that the first factor favored Connectix.

10 Even if the SSO of the Java APIs was copyrightable—which it is not—a reasonable jury
 11 could find that the purpose and character of Google's use of the SSO weighed towards fair use for
 12 the same general reasons the Ninth Circuit found that the first fair use factor favored Connectix.
 13 Although the quasi-commercial aspects of Google's use would not be indirect in the same sense
 14 as in *Sony v. Connectix*, *see* 203 F.3d at 607, the commercial aspects of Google's use are
 15 nonetheless still indirect, because Google does not directly generate revenue by selling or
 16 licensing the SSO of the API packages or the API packages themselves. Instead, Google
 17 generates revenue from Android indirectly, mostly by way of revenue received on advertisements
 18 that appear on Android phones, which is the same way that Google generally generates most of its
 19 revenue from any platform, mobile or otherwise. *See* RT 1458:12-16 (Schmidt). And, as was the
 20 case in *Sony v. Connectix*, Google's use is for the purpose of achieving compatibility—in this
 21 case compatibility with the APIs in the 37 packages at issue. RT 1782:6-17 (Bornstein); RT
 22 2183:6-11 (Astrachan); RT 803:9-20 (Bloch). Weighing these facts together, a reasonable jury
 23 could find that the first statutory factor favors of finding a fair use.

24 **3. A reasonable jury could find that the nature of the SSO of the 37 API**
 25 **packages is highly functional.**

26 The second fair use factor, the nature of the copyrighted work, "reflects the fact that not
 27 all copyrighted works are entitled to the same level of protection." *Sega Enters. Ltd. v.*
 28 *Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992). Of particular note here, the Copyright Act

1 does not protect “functional or factual aspects of the work.” *Id.* (citing 17 U.S.C. § 102(b)).
 2 Works having “strong functional elements” are entitled to less protection than, for example,
 3 works of fiction. *Id.* (citing *Baker v. Selden*, 101 U.S. 99, 104 (1879)).

4 The testimony at trial established that the SSO of the 37 API packages is functional. *See*,
 5 *e.g.*, Dkt. 1047, Findings of Fact 8-9, 14-15 (citing RT 772:17-24, 773:14-16 (Bloch), RT 289:8-
 6 9, 290:8-12 (Ellison), RT 364:3-10 (Kurian), RT 1959:12-1960:18 (Schwartz), RT 784:9-21
 7 (Bloch), RT 1304:5-20 (Mitchell, TX 3542, Mitchell Depo. at 120:18-24, 121:1-10), RT 746:24-
 8 747:9, 747:25-748:6 (Bloch)); *see also* Dkt. 1202 at 4:7-9 (finding that “[t]he overall name tree,
 9 of course has creative elements but it is also a precise command structure – a utilitarian and
 10 functional set of symbols, each to carry out a pre-assigned function.”). A reasonable jury would
 11 be entitled to find what this court has already found: the overall SSO of the Java API packages is
 12 “a utilitarian and functional set of symbols, each to carry out a pre-assigned function.” Dkt. 1202
 13 at 4:7-9. Therefore, a reasonable jury could conclude that this fact outweighs any creativity in the
 14 process of designing the APIs, and that this factor therefore favors a finding of fair use.

15 **4. A reasonable jury could find that Google used only so much of the**
 16 **J2SE API packages as is necessary for compatibility.**

17 The third fair use factor considers the amount and substantiality of the portion used in
 18 relation to the work as a whole. Oracle’s infringement claim is based on the use of the SSO of
 19 only 37 of the 166 API packages in J2SE. RT 597:18-19 (Reinhold) (166 API packages in J2SE
 20 5.0); TX 1072 (37 accused API packages). Those method declarations for those packages
 21 account for just 7,000 lines of code out of the 2,800,000 in the J2SE 5.0 API. RT 2185:10-14,
 22 2190:24-2192:7 (Astrachan). The Android *implementing* code for those API packages is different
 23 than the J2SE *implementing* code. RT 2182:25-2183:1 (Astrachan) (“The implementation code
 24 in Android is completely different than the implementation code in Java.”). Based on these facts
 25 alone, a reasonable jury could find that the third factor favors Google.

26 Moreover, even if a jury found that the portion of the work that Google used is significant,
 27 “[i]f the secondary user only copies as much as is necessary for his or her intended use, then this
 28 factor will not weigh against him or her.” *Kelly v. Arriba Soft Corp.*, 336 F.3d 811, 820-21 (9th

1 Cir. 2003). Here, by using only the SSO and not the implementing code, and by limiting its use
 2 of the SSO to the API packages that developers are most likely to expect to be able to use to write
 3 programs in the Java language for a smartphone platform, Google used only so much of Oracle's
 4 work as was necessary to achieve compatibility as to basic functionalities. *See* RT 2196:7-
 5 2202:11 (Astrachan) (all 37 API packages at issue provide basic functionalities needed to make
 6 practical use of the Java language and are expected by developers). At a minimum, then, a
 7 reasonable jury could find that the third factor weighs neither for nor against fair use. *See Kelly*,
 8 336 F.3d at 821 (copying of entire copyrighted images did not weigh against fair use where "[i]t
 9 was necessary for Arriba to copy the entire image to allow users to recognize the image and
 10 decide whether to pursue more information about the image or the originating web site").

11 **5. A reasonable jury could find that there has been no adverse impact on**
 12 **the actual or potential market for Oracle's work.**

13 "Whereas a work that merely supplants or supersedes another is likely to cause a
 14 substantially adverse impact on the potential market of the original, *a transformative work is less*
 15 *likely to do so.*" *Sony v. Connectix*, 203 F.3d at 607 (citing *Campbell*, 510 U.S. at 591) (emphasis
 16 added). "No 'presumption' or inference of market harm that might find support in *Sony [Corp. of*
 17 *Am. v. Universal City Studios, Inc.*, 464 U.S. 417 (1984),] is applicable to a case involving
 18 something beyond mere duplication for commercial purposes." *Campbell*, 510 U.S. at 591. At
 19 most, such a presumption applies to "verbatim copying of the original in its entirety for
 20 commercial purposes" *Id.* But where the defendant's "use is transformative, market
 21 substitution is at least less certain, and market harm may not be so readily inferred." *Id.* The
 22 Ninth Circuit's analysis in *Sony v. Connectix* is instructive:

23 The district court found that "[t]o the extent that such a substitution [of Connectix's
 24 Virtual Game Station for Sony PlayStation console] occurs, Sony will lose console
 25 sales and profits." Order at 19. We recognize that this may be so. But **because the**
 26 **Virtual Game Station is transformative, and does not merely supplant the**
 27 **PlayStation console, the Virtual Game Station is a legitimate competitor in the**
 28 **market for platforms on which Sony and Sony-licensed games can be played.**
See Sega, 977 F.2d at 1522-23. For this reason, some economic loss by Sony as a
 result of this competition does not compel a finding of no fair use. **Sony**
understandably seeks control over the market for devices that play games Sony
produces or licenses. The copyright law, however, does not confer such a
monopoly. *See id.* at 1523-24 ("[A]n attempt to monopolize the market by making it
 impossible for others to compete runs counter to the statutory purpose of promoting

creative expression and cannot constitute a strong equitable basis for resisting the invocation of the fair use doctrine.”). **This factor favors Connectix.**

203 F.3d at 607-08 (emphases added). As discussed above, a reasonable jury could conclude that Google’s use is sufficiently transformative, and that Android is a “legitimate competitor” in the market for platforms that implement the 37 API packages, not just the manufacturer of a product that supplants those API packages. The record suggests that Oracle seeks to control the market for software implementing the API packages at issue, just as Sony sought to control the market for devices that play Playstation games. *See, e.g.*, RT 374:21-375:5 (Kurian). However, just as copyright law does not give Sony a monopoly over the market for devices that play games Sony produces or licenses, copyright law does not give Oracle a monopoly over the market for software that implements the 37 API packages. Because Google’s use is transformative, the fourth factor favors Google, just as the fourth factor favored Connectix in *Sony v. Connectix*.

Moreover, a reasonable jury could reject Oracle’s evidence of market harm based on alleged fragmentation. First, according to a report Oracle Corporation prepared and submitted to the European Community in connection with its acquisition of Sun, in 2009 vendors already had and would continue to “‘fragment’ Java as a programming language and environment for developers.” TX 2237 at 13 (¶ 15); RT 572:6-20 (Screven). Indeed, Oracle maintains numerous different versions and profiles of the Java platform, with different sets of APIs for each, and code written for one version or profile will not necessarily run on a different version or profile of the Java platform. *See* RT 719:12-725:6 (Reinhold); *see also* TX 3508 at 3 (“Fragmented between Java SE and Java ME, and between Java ME mobile and TV and within mobile and TV.”). Dr. Reinhold testified, however, that the different sets of APIs available on different versions of the Java platform are not a problem:

Write once, run anywhere was never a promise that if you wrote code for one Java platform that it would automatically/magically work on another.

The write once, run anywhere promise is relative to a specific one of the Java platforms. If you write an application that uses Java SE 5, then you can run it on Sun’s implementation, on Oracle’s implementation, on IBM’s implementation, and on others.

Will that same code run on a particular configuration of Java ME? Well, it depends. It might. It might not. It depends which APIs it uses.

RT 725:10-20 (emphasis added); *see also* RT 563:10-564:1 (Screven) (different versions of Java are not “forks” because they are “each editions of Java specifically designed for a particular purpose”). Thus, Oracle witnesses testified that “fragmentation” is harmful only if it occurs *within* a platform. Indeed, Oracle’s counsel specifically argued to the jury that differences *between* platforms are not harmful. RT 930:13-19. A reasonable jury could find that Android could not “fragment” J2SE, because it is a separate platform designed for a particular purpose distinct from that for which J2SE was designed.

Finally, the jury is entitled to rely on evidence at trial that supported a finding that Android *helped* Oracle’s Java business rather than harming it. Oracle’s Java business continues to grow at a double-digit rate. TX 573 at 5; RT 1925:23-1927:6 (Rizvi) (Java platforms business was growing 13% year over year as of 2010); TX 3532 (Rizvi Depo. at 229:13-229:21) (played at RT 1927:14-18) (Java platforms business continues to grow at a rate of approximately 10%). Jonathan Schwartz testified that Android was helping Sun’s Java business. RT 1992:2-19. Had Google taken a *different* route and *not* implemented the Java language and APIs, Mr. Schwartz testified that this “would have been horrible for Sun’s business.” RT 1992:16-19 (Rizvi).

Thus, on the trial record, a reasonable jury could find that Android is transformative, that Oracle’s alleged evidence of “fragmentation” is contradicted by the testimony of its own witnesses, and that far from harming the market for Oracle’s Java platforms, Android helped that market. Thus, a reasonable jury could find, as the Ninth Circuit did in *Sony v. Connectix*, that the fourth factor favors fair use. Weighing all four factors statutory factors together, a reasonable jury could therefore find that Google made a fair use of the SSO of the 37 Java API packages.

E. Oracle is not entitled to judgment as a matter of law on copyrightability.

1. Oracle’s motion is improper.

Oracle’s motion for judgment as a matter of law on copyrightability is improper because Rule 50 does not apply to issues decided by the Court. *Granite State, supra*, 76 F.3d at 1030-31. The jury did not decide copyrightability, which was addressed by the Court as a matter of law.

In order to seek the relief Oracle now seeks, it would have had to file a motion for amended or additional findings under Fed. R. Civ. P. 52(b), which it did not do. Moreover, even

1 if Oracle had filed a Rule 52(b) motion, the motion would be improper. “The purpose of post-
 2 judgment motions under Rules 52(b) and 59 is to give the district court an opportunity to correct
 3 manifest errors of law or fact at trial, allow the parties to present newly discovered evidence, take
 4 additional testimony, make additional findings, or take other action in the interests of justice.”
 5 *Sheldon L. Pollack Corp. v. Universal Health Servs. Inc.*, 1990 WL 194282 at *1 (9th Cir. Dec. 6,
 6 1990) (unpublished); *see also* Rutter Group Prac. Guide Fed. Civ. Trials & Ev. Ch. 17-E at
 7 17:152. ““This is not to say, however, that a motion to amend should be employed to introduce
 8 evidence that was available at trial but was not proffered, to relitigate old issues, to advance new
 9 theories, or to secure a rehearing on the merits.”” *Pollack*, 1990 WL 194282 at *1 (quoting
 10 *Fontenot v. Mesa Petroleum Co.*, 791 F.2d 1207, 1219 (5th Cir.1986)) (emphasis added); *see also*
 11 *ATS Prods. Inc. v. Ghiorso*, No. C10-4880, 2012 WL 1067547 at *1 (N.D. Cal. Mar. 28, 2012)
 12 (“Motions under Rule 52(b) are designed to correct findings of fact which are central to the
 13 ultimate decision; the Rule is not intended to serve as a vehicle for a rehearing.”). Oracle’s
 14 motion under Rule 50(b) should be denied because it is procedurally improper.

15 **2. The Court correctly concluded that the SSO of the API packages is not**
 16 **copyrightable.**

17 Oracle’s motion also fails on the merits. The parties have briefed the issue of
 18 copyrightability extensively. To the extent that the Court accepts Oracle’s incorporation by
 19 reference into its motion for judgment as a matter of law of hundreds of pages of prior
 20 copyrightability filings, Google incorporates by reference its own prior copyrightability filings.
 21 *See* Dkts. 260, 368, 601, 778, 823, 831, 852, 860, 887, 897, 898, 955, 993, 1047, 1079, 1116,
 22 1137, 1192 and 1198.

23 The Court correctly found that by including in the Android API packages the SSO of the
 24 APIs in the 37 J2SE packages at issue, “[c]ode already written in the Java language would, to this
 25 extent, run on Android and thus achieve a degree of interoperability.” Order re Copyrightability
 26 [Dkt. 1202] at 6:17-18. In fact, “the rules of Java dictate the precise form of certain necessary
 27 lines of code called declarations, whose precise and necessary form explains why Android and
 28 Java *must be* identical when it comes to those particular lines of code.” *Id.* at 7:16-19 (emphasis

1 in original); *see also id.* at 35:20-22. Altering the SSO of the APIs would prevent the Android
 2 versions of those APIs from being interoperable with the J2SE versions. *Id.* at 12:19-24. These
 3 findings were amply supported by the trial record. *See, e.g.,* RT 2286:17-2287:8, 2292:25-
 4 2293:14 (Mitchell); RT 1787:20-1788:4 (Bornstein); RT 2168:1-2172:11, 2183:2-20 (Astrachan).

5 Thus, for the 37 packages, Android’s method and interface headers, and field declarations,
 6 had to be *exactly* the same, otherwise by definition they would have implemented *different*
 7 methods, interfaces and fields. *See* Dkt. 1202 at 35:21-22. And “when there is only one way to
 8 write something, the merger doctrine bars anyone from claiming exclusive copyright ownership
 9 of that expression.” *Id.* at 35:23-25; *see also Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446
 10 F.2d 738, 742 (9th Cir. 1971); *Baker v. Selden*, 101 U.S. 99, 104 (1879); *Allen v. Academic*
 11 *Games League of Am., Inc.*, 89 F.3d 614, 617-18 (9th Cir. 1996).

12 In the alternative, although the Court did not so hold, the trial record supports the
 13 conclusion that the particular SSO of the 37 API packages is unprotected under the *scenes a faire*
 14 doctrine, because among Java language developers they are now “commonplace expressions” that
 15 are “indispensable and naturally associated” with the functions for which they are used. *Swirsky*
 16 *v. Carey*, 376 F.3d 841, 850 (9th Cir. 2004); RT 762:13-763:13 (Bloch); RT 961:13-962:3
 17 (Swetland); RT 1018:4-23 (Morrill); RT 1769:11-17, 1782:6-17 (Bornstein); RT 2169:25-
 18 2170:13, 2195:10-2201:17; 2202:6-11, 2203:11-15 (Astrachan); RT 364:17-21 (Kurian); RT
 19 519:16-520:6 (Screven); RT 2289:16-2290:3 (Mitchell); *see also Computer Assocs. Int’l, Inc. v.*
 20 *Altai, Inc.*, 982 F.2d 693, 707-10 (2d Cir. 1992); *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9
 21 F.3d 823, 838 (10th Cir. 1993).

22 Contrary to Oracle’s suggestion, nothing in the Court’s prior order regarding Google’s
 23 motion for summary judgment precludes relief based on the merger and *scenes a faire* doctrines.
 24 The Court declined to grant summary judgment based on these doctrines because it concluded
 25 that Google’s assertion of these doctrines was not adequately supported *based on the summary*
 26 *judgment record*. Dkt. 433 at 9:5-22. The trial record, however, went well beyond the summary
 27 judgment record. *See supra*. Moreover, although the Court stated that “[i]f Google believes, for
 28 example, that a particular method declaration is a *scene a faire* or is they only possible way to

1 express a given function, then Google should provide evidence and argument supporting its views
 2 as to that method declaration,” the Court did not hold that Google was *required* to proceed by this
 3 method of proof, which the Court expressly labeled an “example.” Dkt. 433 at 9:12-14.

4 Although the Court held that *based on the summary judgment record*, Google was not entitled to
 5 categorical rulings with respect to these doctrines, it did not hold that Google was foreclosed from
 6 offering further proof at trial that would support categorical rulings.

7 Finally, the Court correctly found that “[i]n order for at least some [preexisting Java
 8 language code written by third party developers] to run on Android, Google was required to
 9 provide the same *java.package.Class.method()* command system using the same names with the
 10 same ‘taxonomy’ and with the same functional specifications.” Order re Copyrightability [Dkt.
 11 1202] at 38:9-12 (emphasis in original). As the Court further correctly found, “Google replicated
 12 what was necessary to achieve a degree of interoperability—but no more, taking care, as said
 13 before, to provide its own implementations.” *Id.* at 38:12-13.

14 Thus, the SSO at issue is functionally required for compatibility, and therefore not
 15 copyrightable. *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992); 17
 16 U.S.C. § 102(b). Oracle incorrectly characterizes this statement as meaningless, arguing that all
 17 computer programs are functional. Oracle’s argument misses the point; the issue at hand relates
 18 to only the SSO of the API packages, not the implementing source code that is the portion of the
 19 packages most analogous to a computer program. The source code may be copyrightable and
 20 may not be required for *compatibility* with the APIs in the 37 packages—as is readily apparent
 21 from the fact that the Android source code implementing those APIs is completely different from
 22 the J2SE implementing source code.

23 Oracle’s argument that Android is, in some other respects, incompatible with J2SE is
 24 beside the point. First, the Ninth Circuit has not held that a defendant’s software must be
 25 compatible with the entirety of a plaintiff’s *platform* in order for section 102(b) to apply. To the
 26 contrary, Connectix implemented only 137 of the Sony Playstation BIOS’s 242 functions,³ and

27 ³ See Connectix’s Opening Appellate Br. at 18, *available at* 1999 WL 33623860 (9th Cir. May
 28 27, 1999).

the Ninth Circuit nonetheless stated that Connectix’s Virtual Game Station software “itself infringe[d] no copyright.” *Sony Comp. Entm’t, Inc. v. Connectix Corp.*, 203 F.3d 596, 608 n.11 (9th Cir. 2000); *see also* Dkt. 1202 at 39.

The merger and *scenes a faire* doctrines, and three formulations of section 102(b)—functional requirements for compatibility, methods of operation, and systems—all independently support the Court’s conclusion that the SSO of the 37 API packages is not copyrightable. To the extent that the Court reaches the merits of Oracle’s motion, it should be denied.

F. Oracle is not entitled to judgment as a matter of law on derivative work.

Oracle claims that the Android source code for the 37 API packages was derived from Oracle’s copyrighted API specifications. Material only qualifies as an unlawful derivative work if it is derived from material that is itself copyrightable. *Mirage Editions v. Albuquerque A.R.T. Co.*, 856 F.2d 1341, 1343 (9th Cir. 1988). Because the material from which the Android source code allegedly is derived is not copyrightable, Oracle’s derivative work claim fails.

The Android source code is not even arguably “derived” from anything that is copyrightable. The literal aspects of the APIs, such as the method headers, are not copyrightable because they are functional requirements for compability. *See* Part I.G, *supra*. Oracle’s only other argument is that Google’s *implementing source code* is an unlawful derivative work of *Oracle’s English-language descriptions* because Google’s source code *does the things that the English descriptions describe*. *See* Dkt. 859 at 10 (Oracle claimed infringement based on “Google’s creation of derivative works from the English-language descriptions of the elements in the API specifications”). That is nothing but an assertion that *Google’s expression* infringes *Oracle’s ideas*, which is a “classic case of trying to lay claim to the ownership of an idea.” RT 1869:15-16 (Court); 17 U.S.C. § 102(b). The specifications “explain in detail what the module is supposed to accomplish,” and writing implementing code that *does* what the specifications *explain* is like “creative writing.” RT 1869:18-21 (Court).⁴ Thus, the Android source code is not

⁴ *See also* RT 1368:25-1369:1 (Court) (Oracle’s derivative work argument “just seems to me to be invalid under the basic tenets of copyright law”); RT 1375:22-24 (Court) (Oracle’s derivative work claim doesn’t “add[] anything, except violating the principle of you can’t get a monopoly and ownership over an idea”); RT 2434:13-2435:16 (Court) (rejecting derivative work theory).

“derived” from any Oracle material that is protected by copyright.

The cases Oracle cites are inapposite, because they assume that the accused work is derived from *copyrightable expression* from the original work. For example, in *Micro Star v. Formgen Inc.*, the court concluded that the defendant’s work was derived from the plaintiff’s “story,” and that the defendant’s work was an unauthorized “sequel.” 154 F.3d 1107, 1112 (9th Cir. 1998). Here, the Court found that the literal elements of the APIs are not copyrighted, and that the English-language descriptions explain an uncopyrightable *idea*. Because none of the material from which the Android source code allegedly is derived is copyrightable, the Android source code is not an unlawful derivative work. *Mirage Editions*, 856 F.2d at 1343.

II. THE JURY PROPERLY FOUND THAT GOOGLE DID NOT INFRINGE THE ASSERTED CLAIMS OF THE 104 PATENT.

Under the Court’s claim construction, no reasonable jury could find that Android infringes Claims 11, 27, 29, 39, 40, or 41 of U.S. Patent No. RE38,104 (the ’104 patent).

A. Dalvik bytecode instructions only contain indexes, which are not “symbolic references” because they refer to numeric memory locations.

Oracle now retreats from the argument that references can be both symbolic and numeric. Instead, Oracle focuses on a related argument that the “data” being referenced is “the value of the data from the instance of an object.” Mot. at 18. This theory suffers the same flaws.

1. Field indexes are numeric references, not symbolic references.

Dalvik bytecode instructions consist of an opcode and an operand or “argument.” TX 737; RT 3221:8-10, 3234:4-7 (McFadden); RT 3590:23-3591:25 (Bornstein); RT 3844:14-3855:12, 3925:4-18 (August). The accused operands, moreover, are indexes that take the form “field@CCCC” and provide references to locations in tables. *See* Mot. at 17, 18; TX 46.106; TX 735; TX 737; RT 3732:15-19, 3736:16-23, 3755:8-9, 3765:9-12 (McFadden); 3858:5-12, 3858:21-359:11, 3918:13-23, 3923:20-24, 3925:19-3926-9 (August). Oracle’s own expert, Dr. Mitchell, agrees. RT 3488:6-8, 3488:19-23, 3489:10-12 (Dr. Mitchell); RT 3496:12-3497:6. These indexes refer to locations in the constant pool tables, which are data in the .dex file; for example, “field@0001” refers to the entry 1 in the Field IDs table. *See* Mot. at 17.

As in its original JMOL motion, Oracle focuses on the IGET instruction in Dalvik

bytecode. *See id.* Mot. at 18. It argues that the IGET instruction corresponds to the “LOAD ‘y’” instruction in Figure 1B of the ’104 patent. *Id.* But that is the wrong example. As Oracle notes, “[t]he IGET instruction contains three operands—vA, vB, and field@CCCC—where the third operand field@CCCC is the field index.” *Id.* at 3 (citing TX 735 at 6; RT 3221:8-10 (McFadden)). That “field index in the IGET instruction identifies the field from which the data is to be obtained by IGET.” [*Id.*] The IGET instruction therefore actually corresponds to the “LOAD 2” instruction depicted in Figure 1A of the ’104 patent. That instruction directs the computer to go to Slot 2 in the table, much like the IGET instruction 52 01 instructs the computer to go to entry 01 in the Field ID table. Oracle concedes this is how the IGET instruction works. *Id.* at 18. In short, the operands for IGET instructions in Dalvik bytecode—the indexes—therefore qualify as numeric references rather than symbolic references. Those indexes refer to data by a numeric memory location—entry 01 in the Field IDs table. They therefore are numeric references that do not meet the Court’s construction of “symbolic references.”

B. The Court’s claim construction applies to data in the constant pool tables as well as the actual field data in an instance object.

Oracle admits that indexes in the instruction stream indicate the location of information in the constant pool tables. Oracle nonetheless contends that the indexes in Dalvik bytecode instructions do not qualify as “numeric” references because they do not identify “the value of the data from the instance of an object.” Mot. at 18-19. Put another way, according to Oracle, an index “identifies data by a name other than the numeric memory location of *the data*,” and therefore cannot meet the Court’s construction of a “symbolic reference.” Mot. at 17. There is no basis for Oracle’s narrow interpretation of the term “data.”

The Court’s claim construction distinguishes between using names to represent data (i.e., symbolic references) and using numeric memory locations. Dkt. No. 137 at 22. Dr. Mitchell agrees. RT 3480:12-15. There is no limitation regarding the type of data being referenced. Yet Oracle argues “[t]hat data from an instance object is the ‘data’ that the claimed symbolic reference refers to, and not some other data,” Mot. at 19. The jury could reasonably reject such a narrow reading of the term “data,” particularly when Dr. August testified that information

1 in the constant pool tables is identified by location using an index value in the Dalvik bytecode
 2 instructions; there is no search performed to determine the location of the data. RT 3865:11-20.

3 As Dr. August further explained, the information in the constant pool is data. RT
 4 3955:22-24. And the Court’s claim construction simply distinguishes between a reference to
 5 data—any data—by a name rather than location. *See* Claim Construction Order (Dkt. 137) at 20-
 6 22; RT 3480:12-15 (Mitchell). The construction included no limitations or requirements
 7 regarding the type of data being referenced. In any event, the theory Oracle posits—that “data”
 8 must refer to some ultimate value in an instance object—is disproved by testimony that such
 9 “data” may itself be a reference (an index) to yet other data. RT 4002:14-16; 4003:11-17.

10 By claiming that only the value of the actual field data in an instance object qualifies as
 11 “data” under the Court’s construction (whereas the data in constant pool tables does not), Oracle
 12 inherently claims that references can be *both* symbolic and numeric. That is, depending on which
 13 “data” one uses as the relative end point, upstream references would all be deemed “symbolic”,
 14 regardless of whether they were simple indexes to the location of other data. The Court has
 15 already—correctly—rejected this argument in light of the compelling evidence to the contrary in
 16 the patent specification. *See, e.g.*, TX 4015 (’104 Patent) at col. 2, ll 41-55.

17 **C. Indexes are not symbolic references because they are not resolved.**

18 Oracle contends that field indices are “resolved” into pointers, making them “symbolic
 19 references,” and points to Mr. McFadden’s testimony for support. Mot. at 19-20. But Mr.
 20 McFadden testified that it does not make sense to “resolve” field indices, which tell the computer
 21 where to go in memory; rather, resolution is done when the computer doesn’t know where to go.
 22 RT 3650:18-3651:1. The document Oracle cites does not suggest otherwise. To the contrary, it
 23 explains the existence of a chain of *indexes*, which indexes lead to *strings*, which strings are used
 24 by the *resolver* determine the instance object. TX 46.12 at 1. It is these strings and not the
 25 indexes that are “resolved” for program to determine where to go in memory.

26 In fact, indexes in Dalvik bytecode instructions always point to a numeric memory
 27 location of data, both before and after resolution of the symbolic references in the .dex file string
 28 data. In Resolve.c, the instruction stream does not change; the index field@CCCC is used both

before and after resolution to look at entries in the Resolved Fields Table and the Field ID table. RT 3636:13-3637:20 (McFadden). In dexopt, the bytecode instruction operand is simply changed from one numeric reference to another numeric reference. RT 3746:22-3747:14 (McFadden); 3933:5-3934:19 (August). Specifically, the index value “field@CCCC” is replaced with another numeric memory location—the offset “fieldoff@CCCC.” RT 3746:22-3747:14 (McFadden); 3933:5-3934:19 (August); TX 737 at 3; TX 46.106; *see also* TX 737 at 1; TX 739.

D. Dexopt does not infringe the ’104 patent.

Oracle’s request for judgment as a matter of law on infringement of claims 27 and 29 should also be denied. Oracle’s claim that the Android dexopt tool resolves symbolic references dynamically rather than statically rests on Dr. Mitchell’s erroneous opinion that an optimization (e.g., resolution) based on information *about* a runtime environment is “dynamic.” But there is a wealth of documentary evidence and fact and expert testimony that proves dexopt is a static optimization. RT 3730:16-22 (McFadden); RT 3940:17-20 (August); RT 3595:21-24 (Bornstein); TX 32 at 35, TX 816 at 24:05; TX 735 (defining opcodes that “are reasonable candidates for static linking”); TX 737 (defining “statically linked” instruction formats); TX 739 (dexopt performs optimizations that “can be inferred statically”). Oracle’s reliance on TX 1094 remains specious. As is clear, it was an incomplete cut-and-paste job from TX 739 (also TX 105). And the copied reference to runtime information, as explained by Andy McFadden, does not refer to optimizations performed by dexopt. *See* RT 3734:20-3735:7. Accordingly, no reasonable jury could conclude that dexopt infringes claims 27 and 29 of the ’104 patent.

In any event, the jury found that the other asserted claims were not infringed. At the least, there was insufficient proof that Dalvik bytecode instructions contain symbolic references. As such, there is an independent basis for affirming the verdict on Claims 27 and 29.

III. THE JURY PROPERLY FOUND THAT THE ANDROID DX TOOL DID NOT INFRINGE THE ASSERTED CLAIMS OF THE ’520 PATENT.

Oracle argues yet again that the “ordinary meaning” of “simulating execution” compels a finding of infringement of the ’520 patent. Oracle might believe that the third time will be the charm; however, its argument, already rejected both by the jury and by the Court, fares no better

1 this time around. The root of Oracle’s argument is that simulating execution of bytecode requires
 2 no more than that “bytecode instructions are examined without being executed.” Mot. at 21:26.
 3 Yet, as the Court correctly observed in its Order denying Oracle’s first Rule 50 motion, Oracle
 4 never sought to construe the term “simulating execution.” Dkt. 1201 at 10:10-12. Instead, Oracle
 5 took the position that the term should be understood in accordance with its plain meaning.
 6 Professor Parr’s testimony, which the jury was entitled to credit, was that the sort of pattern
 7 matching performed by the Android dx tool does not fall within the plain meaning that one skilled
 8 in the art would give “simulating execution.” Dr. Mitchell apparently disagreed, but the jury was
 9 not required to accept his opinion. Dkt. 1201 at 11:13-17. Although Oracle plainly does not
 10 agree with the jury’s decision to credit Professor Parr’s opinion, it cannot now insist on a different
 11 construction of the term as matter of law after taking the position throughout this litigation that
 12 there was no need to construe the term in the first place.

13 Leaving aside the untimeliness of Oracle’s request for its desired construction of the term
 14 “simulating execution,” the arguments Oracle presents for why the dx tool allegedly infringes are
 15 flawed. In its two-page discussion of the issue, Oracle references not once, but twice, the source
 16 code comments saying that the simulator.java class knows how to simulate the effects of
 17 executing bytecode. Mot. at 21:16-18, 21:21-23. Yet those comments are consistent with the
 18 manner in which Google’s witnesses testified that the dx tool works when they explained why it
 19 did not infringe: The testimony at trial by both Mr. Bornstein and Professor Parr is that the
 20 simulator.java class does simulate execution of bytecode—but that it does so only for purposes
 21 *other than* identifying the static initialization of an array. RT 3589:3-3590:6 (Bornstein); RT
 22 3798:3-16 (Parr)]. Moreover, both parties’ experts agreed that the code that identifies the static
 23 initialization of arrays is not found in the simulator.java class; instead, it is found in the separate
 24 BytecodeArray class. RT 3799:14-3800:7 (Parr); 3519:19-3520:3 (Mitchell); 4061:9-4062:3
 25 (Mitchell). That separate class identifies static initialization of arrays through pattern matching
 26 rather than through simulating execution. RT 3799:14-3801:18 (Parr). Indeed, Oracle concedes
 27 that the dx tool does perform pattern matching, Mot. at 22:17-19, while arguing that the claims do
 28 not exclude pattern matching from their ambit. But it was Oracle’s burden to prove infringement

of the claims as properly construed, not Google's task to prove that the dx tool does not infringe.⁵

In sum, Oracle's assertions that the Court limited the claims of the '520 patent to a disclosed embodiment, Mot. at 22:22-23, or that Google has read additional limitations into the claims, Mot. at 22:6-7, are incorrect. The '520 patent expressly says that methods "consistent with the present invention" yield "an improved system for initializing static arrays in the Java™ programming environment." TX 4011 at col. 3, 43-45. Even Dr. Mitchell admitted that "in a Java bytecode system, instructions operate by pushing and popping and replacing values from the top of an operand stack." RT 4058:3-6 (Mitchell). Accordingly, the Court's understanding, consistent with Professor Parr's testimony, that "simulating execution" within the context of the '520 patent means "the simulation of actual Java virtual machine operations," (Dkt. 1201 at 10:17-18), is a reflection of the plain meaning of the claim language, not a limitation of the claims to a particular embodiment. Oracle's argument that the claims cover more than what one skilled in the art would understand simulating execution by this sort of virtual machine to mean is odd, for Oracle itself has throughout this litigation insisted that its patents are "Java-related" patents. E.g., Oracle's Trial Brief (Dkt. No. 538) at 1.

IV. ORACLE IS NOT ENTITLED TO JUDGMENT AS A MATTER OF LAW ON GOOGLE'S EQUITABLE DEFENSES

A. Oracle's motion is improper.

The Court addressed Google's equitable defenses in its Findings of Fact and Conclusions

⁵ Oracle also presses a claim differentiation argument that was neither found in Dr. Mitchell's expert report nor testified to by any witness at trial. Mot. at 22:12-13. The time to make such an argument, if at all, would have been during the Markman process, or at the latest as part of the jury charge conference. It is far too late to raise it now. However, had Oracle attempted to make the argument in a timely fashion, Google would have shown why it has no merit. Unasserted claim 3 of the '520 patent, upon which Oracle seeks to rely for a claim differentiation argument, requires that the "play executing step" of claim 1 include within it three additional steps: the steps of allocating a stack, reading a bytecode that manipulates the stack, and performing stack manipulation on the stack. Because the dependent claim requires that all three steps be part of the play executing step, a method that allocated a stack as part of the second step of claim 1 ("receiving the class into a preloader") but performed the other two steps (reading a bytecode and manipulating the stack) as part of the third ("play executing") step of claim 1 would not infringe claim 3, and would still infringe claim 1. The point of difference between the claims is thus not merely manipulation of a stack. Thus, even if claim differentiation were an iron-clad rule—and it is not, *see Eon-Net LP v. Flagstar Bancorp*, 653 F.3d 1314, 1323 (Fed. Cir. 2011)—claim differentiation does not prevent claim 1 from covering stack manipulation.

of Law. Dkt. 1203 at 3. Oracle’s motion for judgment as a matter of law on Google’s equitable defenses of equitable estoppel and laches is improper because, as discussed above in Sections I.A. and I.E.1, Rule 50 does not apply to issues decided by the Court. *Granite State*, 76 F.3d at 1030-31 (9th Cir. 1996). If Oracle wishes to challenge the Court’s ruling on those defenses, Oracle must file a motion for amended or additional findings under Fed. R. Civ. P. 52(b).

B. Oracle’s motion is moot.

Even if Oracle’s motion was proper, Oracle would not be entitled to the relief it seeks. Oracle asks the Court to grant judgment in its favor on two defenses—equitable estoppel and laches—that had no effect on the final outcome of this case. The Court has held that these defenses “can be revived” by Google if the case is remanded. [Dkt. 1203]. Re-litigating these issues now serves no purpose. *ATS Prods. Inc.*, 2012 WL 1067547 at *1 (N.D. Cal. March 28, 2012) (“Motions under Rule 52(b) are designed to correct findings of fact which are central to the ultimate decision; the Rule is not intended to serve as a vehicle for a rehearing. Accordingly, Oracle’s motion should be denied as moot. *Id.* (“[A] motion to amend a court’s factual and legal findings is properly denied where the proposed additional facts would not affect the outcome of the case or are immaterial to the court’s conclusions.”)).

C. Oracle’s substantive arguments also lack merit.

There are no grounds for amending the Court’s equitable estoppel or laches findings. All of the arguments and specific evidence in Oracle’s renewed motion appeared either in Oracle’s proposed findings of fact and conclusions of law (Dkt. 1049) or in its earlier JMOL motions (Dkts. 1045 and 1168), and Google therefore incorporates by reference the evidence and argument in its responses to those filings. Dkts. 1047, 1079, 1092 and 1169. Oracle previously failed to persuade the Court that a preemptive finding in Oracle’s favor is warranted on these issues. Dkt. 1203 at 3:14-20. There is no reason that it should prevail now.

1. There is sufficient evidence to support a finding that Oracle’s claims are barred by equitable estoppel.

Even if the Court were to address Google’s estoppel defense on the merits, there would be sufficient evidence to support a finding in Google’s favor, as detailed in Google’s Proposed

Findings of Fact and Conclusions of Law, Dkt. 1047 at 32-33, Google's Responses to Oracle's Proposed Findings of Fact and Conclusions of Law, Dkt. 1079 at 59-62, and Google's Opposition to Oracle's Rule 50(a) Motion, Dkt. 1169 at 14-18, all of which Google incorporates herein by reference. Sun knew about and approved unlicensed, open source implementations of the Java API packages as long as the implementation did not use the Java brand. *See* Dkt. 1047, Findings of Fact 37-52. As early as 2005, Sun knew Google intended to implement Java API packages in Android, and Sun never told Google it needed a license to do so. *See id.*, Findings of Fact 53-61. After Google publicly announced Android, Sun congratulated Google and welcomed Google to the Java community. *See id.*, Findings of Fact 62-72. After Google's announcement of Android and release of the Android SDK, Sun continued to talk with Google and publicly support Android. *See id.*, Findings of Fact 73-81. Google was aware of and relied on Sun's public statements of approval and acts of support for the Android platform. *See id.*, Findings of Fact 82-87. Oracle initially encouraged Android and tried to partner with Google. *See id.*, Findings of Fact 88-92. Accordingly, there is sufficient evidence to support a finding for Google on estoppel.

2. Oracle's claims are barred by laches.

The facts that support Google's equitable estoppel defense also support its laches defense. Dkt. 1047 at 31-32; 1079 at 64-66; 1169 at 16-17. That Sun occasionally tried to convince Google to take a license (in addition to entering into a partnership) for Android—but did not identify or even allude to the copyrights-in-suit or patents-in-suit—does not excuse Oracle's delay in filing suit. Each of the cases Oracle cites for the proposition that negotiations are an excuse for unreasonable delay involve negotiations focused on the specific intellectual property at issue. *See Lucent Techs., Inc. v. Gateway, Inc.*, 580 F. Supp. 2d 1016, 1053 (S.D. Cal. 2008) *aff'd in part, vacated in part on other grounds, remanded*, 580 F.3d 1301 (Fed. Cir. 2009) ("The evidence showed that for several years leading up to the start of the litigation underlying this case, beginning around 1998 through 2002 and 2003, Lucent engaged in efforts to sell a license for the '226 to computer manufacturers Gateway and Dell."); *see also In re Katz Interactive Call Processing Patent Litig.*, 712 F. Supp. 2d 1080, 1110 (C.D. Cal. 2010). Sun and Google never negotiated over the copyrights-in-suit or patents-in-suit, or any specific copyrights or patents for

that matter. RT 3190:25-3192:2, 3181:25-3182:10 (Rubin). Accordingly, those “negotiations” do not excuse Oracle’s delay in filing suit.

V. ALTERNATIVE DEFENSES TO PATENT INFRINGEMENT

Oracle moves for judgment as a matter of law as to Google’s Sixth (patent misuse), Eighth (use by the U.S.) and Nineteenth (unclean hands) defenses, as well as Google’s defense of express license. Though Google believes these defenses have merit, Google chose not to present them at trial and therefore did not oppose Oracle’s Rule 50(a) motion on the defenses. Dkt. 1169 at 18-19. Similarly, Google does not oppose Oracle’s Rule 50(b) motion on those defenses.

VI. ORACLE IS NOT ENTITLED TO A NEW TRIAL

Oracle has failed to show that either the jury’s verdict or the Court’s own rulings are erroneous in any way, let alone that they are “against the weight of the evidence.” Dkt. 1212 at 24. Accordingly, Oracle’s alternative request for a new trial should be denied.

A new trial may be warranted where “the verdict is against the weight of the evidence, [] the damages are excessive, or [] for other reasons, the trial was not fair to the party moving.” *Molski v. M.J. Cable, Inc.*, 481 F.3d 724, 729 (9th Cir. 2007) (quoting *Montgomery Ward & Co. v. Duncan*, 311 U.S. 243, 251 (1940)).” Oracle simply contends that it “is entitled to a new trial for all the same reasons it is entitled to JMOL.” Dkt. 1212 at 25. For all the reasons explained above, Oracle is not entitled to JMOL, and therefore is not entitled to a new trial on any issue.

CONCLUSION

For the all of these reasons, Oracle’s motion should be denied.

Dated: July 5, 2012

KEKER & VAN NEST LLP

By: /s/ Robert A. Van Nest
ROBERT A. VAN NEST

Attorneys for Defendant
GOOGLE INC.